<https://radixweb.com/blog/node-js-vs-other-backend-technologies>

<https://www.cronj.com/blog/how-does-node-js-differ-from-other-frameworks/>

<https://www.geeksforgeeks.org/what-is-pern-stack/>

<https://evrone.ru/blog/articles/react-vs-vue>

https://surf.ru/chto-takoe-middleware/

В качестве СУБД для нашего проекта была выбрана PostgreSQL.

Существует множество стеков технологий с использованием PostgreSQL. В большинстве своем это стеки, который используют PostgreSQL, Express.js и Node.js. На место фронтенд-фреймворка обычно ставят Angular, ReactJS или Vue.js. Ранее были рассмотрены различные фреймворки, в ходе этого исследования мы выяснили, что больше всего для нашего проекта подойдут ReactJS или Vue.js.

**Что мы выбираем для серверной части:**

1. Node.js

Это среда выполнения с открытым кодом для разработки серверной части с помощью JavaScript. Он позволяет работать с бэкендом приложения без использования PHP, Java, Ruby, Python. Node.js комбинирует технологии фронтенд и бэкенд разработки.

Он легок для изучения и эффективен за счет его событийно-ориентированной модели и асинхронности. Это один из лучших вариантов для приложений, которые требуют хорошей масштабируемости.

Если сравнивать Node.js и PHP, то Node.js больше подходит для приложений, которые работают с большими объемами данных. Он может обрабатывать одновременные запросы, не блокируя ни один из них, в то время как PHP этого не может.

Для работы Node.js использует один поток, в отличие от многопоточных систем, таких как PHP, JSP, ASP.NET. Многопоточные системы, как правило, имеют максимальное количество потоков, при превышении этого количества производительно приложения сильно снижается, а также используется больше памяти.

1. Express.js

Express представляет собой популярный веб-фреймворк, написанный на JavaScript и работающий внутри среды исполнения Node.js. Этот модуль освещает некоторые ключевые преимущества этого фреймворка, установку среды разработки и выполнение основных задач веб-разработки и развёртывания.

Он обладает высокой гибкостью, а также прост в изучении.

Express был создан для того, чтобы с легкостью создавать API и веб-приложения, он экономит время на кодирование почти в два раза и при этом делает веб- и мобильные приложения эффективными.

**Особенности Express JS**

* Быстрая разработка на стороне сервера

Возможности node js помогают Express JS экономить много времени.

* Middleware

Middleware - это обработчик запросов, который имеет доступ к циклу "запрос-ответ" приложения. Middleware — связующее программное обеспечение, которое помогает приложению и серверу обмениваться друг с другом запросами. Оно снижает зависимость от API, позволяет не торопиться с обновлением старого бэкенда, снижает нагрузку на мобильный клиент

* Маршрутизация

Маршрутизация в ExpressJS позволяет определить, как приложение реагирует на различные URL-адреса и HTTP-методы. Маршруты могут быть параметризованными и поддерживать разнообразные обработчики запросов. В ExpressJS вы можете определить маршруты с помощью методов, соответствующих HTTP-методам (GET, POST, PUT, DELETE и другие).

* Шаблонизация

Предоставляет шаблонизаторы для создания динамического контента на веб-страницах путем создания HTML-шаблонов на сервере.

* Отладка

Express облегчает отладку, поскольку определяет, где именно находятся ошибки.

**Стек для фронтенда**

Для разработки пользовательского интерфейса будут использоваться HTML, CSS, JavaScript, для связи с бэкендом и базами данных могут использоваться либо фреймворк Vue.js, либо библиотека React

Vue.js подойдет для начинающих разработчиков, так как он прост в изучении, а также для разработки небольших приложений

React имеет преимущество благодаря легкой масштабируемости.

Касательно производительности, развитости сообщества и работы с компонентами эти инструменты очень схожи друг с другом, поэтому для разработки нашего приложения можно использовать любой из них, опираясь на способности бэкенд-разработчика.